EEEEEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFFFFF
EEEEEEEEEEEEE	RRRRRRRRRRR	FFFFFFFFFFFFF
ÉÉÉÉÉÉÉÉÉÉÉÉÉÉ	RRRRRRRRRRR	FFFFFFFFFFFFF
EEE	RRR RRR	FFF
EEE		
	RRR RRR	FFF
EEE	RRR RRR	FFF
EEE	RRR RRR	FFF
EEE	RRR RRR	FFF
ĒĒĒ	RRR RRR	FFF
EEEEEEEEEE	RRRRRRRRRRR	FFFFFFFFFF
EEEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFF
EEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFF
EEE	RRR RRR	FFF

EEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
EEEEEEEEEEEEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
	mm mm	111

\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$

\$\$\$\$\$\$ \$\$\$\$\$\$

\$\$ \$\$ \$\$ \$\$

KK KK KK

KK KK

KK KK

, KK KK KK

\$\$ \$\$ \$\$ \$\$

\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$

\$\$\$\$\$\$ \$\$\$\$\$\$

\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$

• • • •

• • • •

- 5
=
Ζ,
~
τ
J
2
~
À
_
ぇ
1
Ξ
,
<u>-</u>
7
č
7
Ç
ว
C
2
~
ā
2
ぅ
1
Ξ
/
Ξ
7
Ξ
7
Ē
2
C
2
<
2
~

LAE

FUN

T

DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD	QQQQQ QQ QQ QQ QQ QQ QQ QQ QQ QQ QQ QQ	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
		\$

C *

(*

(*

C*

C *

(*

(*

(* (* (* *

C

C++

C

C

C

C C C

C

C

C

C

C

C

C

C

C

C

C

C

0049

0050

0051

0052

0053

0054

0055

0056

0057

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

Author Brian Porter

Creation Date 20-JUL-1981

functional description:

This module is used to display error log entries logged for the 11/7zz IDC. The format of the device specific portion of the record is as follows.

csr
bar
bcr
dar
mpr
ecc1
ecc2
data path number
data path reg (always 0)
final uba map

28-Mar-1984

20-Jun-1983

9-Jun-1983

23-JAN-1982

23-NOV-1981

04-NOV-1981

30-SEP-1981

29-SEP-1981

14-SEP-1981

31-AUG-1981

24-AUG-1981

Added CSR

Added 'device attention'

```
0058
0059
                          previous uba map
        C
        C
0060
                       vec$l_mapreq (from crb)
0061
0062
0063
                 Modified by:
0064
0065
                 V03-003 SAR0217
                                          Sharon A. Reynolds,
0066
                         Changed the call to UCB$L_OWNUIC to ORB$L_OWNER.
0067
0068
                 V03-002 SAR0069
                                          Sharon A. Reynolds,
0069
                         Changed the carriage control in the 'format' statements
0070
                         for use with ERF.
0071
0072
                 v03-001 SAR0046
                                          Sharon A. Reynolds,
0073
                         Removed brief/cryptic support.
0074
0075
                 v02-008 BP0008
                                          Brian Porter,
0076
                         Corrected polarity of 'plug valid' for the r80.
0077
0078
                 v02-007 BP0007
                                          Brian Porter,
0079
                         Minor edit.
0080
0081
                 v02-006 BP0006
                                          Brian Porter,
0082
                         Corrected 'DAR' output error.
0083
                         support.
0084
0085
                 v02-005 BP0005
                                          Brian Porter.
0086
                         Corrected random problems.
0087
0088
                 v02-004 BP0004
                                          Brian Porter,
0089
                         Added 'DAR' decoding functionality.
0090
0091
                 v02-003 BP0003
                                          Brian Porter,
0092
                         Corrected problem in attention logic.
0093
                         functionality.
0094
0095
                 v02-002 BP0002
                                          Brian Porter.
0096
                         Corrected call to calc_map.
0097
0098
                 v02-001 BP0001
                                          Brian Porter,
0099
                         Changed record format to conform to other drivers.
0100
        C * *
0101
        C ---
0102
0103
0104
0105
                 Subroutine DQDISKS (lun)
0106
```

include 'src\$:msghdr.for /nolist' include 'src\$:deverr.for /nolist'

byte

0107 0108

0167

Lun

integer*4 control_status_register integer*4 bus_address_register

DQD

PRO

0

3

Page

VAX-11 FORTRAN V3.4-56

DISK\$VMSMASTER: [ERF.SRC]DQDISKS.FOR: 1

```
DQDISKS
                                                                                 16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
0273
0275
0275
0276
0277
0278
0281
0282
0283
                     integer*4
                                         byte_control_register
                     integer*4
                                         disk_address_register
                     integer*4
                                         multī_purposē_rēgister
                     integer*4
                                         ecc_position_register
                     integer*4
                                         ecc_pattern_register
                     integer*4
                                         data_path_number
                                         data_path_register
                     integer*4
                     integer * 4
                                         final_map_register
                    integer*4
                                         previous_map_register
                    integer*4
                                         vec$l_mapreg
0284
0285
0286
0287
0288
                    equivalence
                                         (emb$i_dv_regsav(0),control_status_register)
                                         (emb$l_dv_regsav(1),bus_address_register)
                    equivalence
                                         (emb$l_dv_regsav(2),byte_control_register)
(emb$l_dv_regsav(3),disk_address_register)
(emb$l_dv_regsav(4),mult1_purpose_register)
                    equivalence
                    equivalence
                    equivalence
0289
0290
0291
0292
0293
                                         (emb$l_dv_regsav(5),ecc_position_register)
                    equivalence
                                         (emb$l_dv_regsav(6),ecc_pattern_register)
                    equivalence
                                         (emb$[_dv_regsav(7),data_path_number)
                    equivalence
                    equivalence
                                         (emb$l_dv_regsav(8),data_path_register)
                                         (emb$l_dv_regsav(9),final_map_register)
(emb$l_dv_regsav(10),previous_map_register)
                    equivalence
0294
0295
                    equivalence
                    equivalence
                                         (emb$l_dv_regsav(11),vec$l_mapreg)
0296
0297
0298
                    character*12
                                         v1csr(0:0)
                                         v1csr(0)
                    data
                                                             /'DRIVE READY*'/
0299
0300
                    character*17
                                         v2csr(6:7)
0301
                                                             /'INTERRUPT ENABLE*'/
                    data
                                         v2csr(6)
0302
                                         v2csr(7)
                                                             /'CONTROLLER READY+'/
                    data
0303
0304
                    character*21
                                         v3csr(10:10)
0305
                    data
                                         v3csr(10)
                                                             /'OPERATION INCOMPLETE*'/
0306
0307
                    character*20
                                         v4csr(13:15)
0308
                                         v4csr(13)
                    data
                                                             /'NON-EXISTENT MEMORY+'/
0309
                    data
                                         v4csr(14)
                                                             /'DRIVE ERROR+'/
0310
0311
                    data
                                         v4csr(15)
                                                             /'COMPOSITE ERROR*'/
0312
                                         v5csr(22:24)
                    character*22
                                         v5csr(22)
v5csr(23)
                    data
                                                             /'R80 SKIP SECTOR ERROR+'/
/'R80 SKIP SECTOR ERROR*'/
                    data
                                         v5csr(24)
                    data
                                                             /'INTERRUPT REQUEST*'/
                    character * 30
                                         v6csr(26:28)
                                         v6csr(26)
v6csr(27)
                                                             /'R80*'/
                    data
                    data
                                                             /'AUTOMATIC SKIP SECTOR INHIBIT*'/
                                         v6csr(28)
                                                             /'TIMEOUT INHIBIT+'/
                    data
                                        v1rl02_mpr(3:5)
v1rl02_mpr(3)
v1rl02_mpr(4)
v1rl02_mpr(5)
                    character*11
                                                             /'BRUSH HOME+'/
                    data
                                                             /'HEADS OUT+'/
                    data
                                                             /'COVER OPEN+'/
                    data
                                        v2rl02_mpr(8:15)
v2rl02_mpr(8)
v2rl02_mpr(9)
                    character + 19
                                                             /'DRIVE SELECT ERROR+'/
                    data
                                                             /'VOLUME CHECK+'/
                    data
```

```
DQDISKS
                                                                                                      16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
                                                                                                                                             VAX-11 FORTRAN V3.4-56
                                                                                                                                                                                                      Page
                                                                                                                                             DISK$VMSMASTER:[ERF.SRC]DQDISKS.FOR:1
                                                   v2rl02_mpr(10)
v2rl02_mpr(11)
v2rl02_mpr(12)
v2rl02_mpr(13)
v2rl02_mpr(14)
v2rl02_mpr(15)
/'WRITE GATE ERROR*'/
                          data
                          data
                                                                             /'SPINDLE ERROR+'/
                          data
                                                                             /'SEEK TIMEOUT+'/
                                                                             /'WRITE LOCK+'/
                          data
                                                                             /'HEAD CURRENT ERROR+'/
                          data
                          data
                                                                             /'WRITE DATE ERROR*'/
                                                   v1r80_mpr(8:13)
v1r80_mpr(8)
v1r80_mpr(9)
v1r80_mpr(10)
v1r80_mpr(11)
v1r80_mpr(12)
v1r80_mpr(13)
                          character*14
                          data
                                                                             /'FAULT+'/
                                                                             /'PLUG VALID+'/
                          data
                                                                             /'SEEK ERROR+'/
                          data
                          data
                                                                             /'ON CYLINDER+'/
                          data
                                                                             /'DRIVE READY+'/
                                                                             /'WRITE PROTECT+'/
                          data
                          integer*4
                                                   compress4
                          integer*4
                                                   compresso
                                                    field
                          integer*4
                                                    idc_command(0:7)
                          character*27
                                                                            /'NO DRIVE OPERATION*'/
/'WRITE CHECK DATA*'/
/'GET STATUS*'/
/'SEEK*'/
                          data
                                                    idc_command(0)
                                                   idc_command(1)
idc_command(2)
idc_command(3)
                          data
                          data
                          data
0354
                                                                             /'READ HEADER+'/
                                                    idc_command(4)
                          data
                                                                             /'WRITE DATA+'/
0355
                                                    idc_command(5)
                          data
0356
0357
                                                                            /'READ DATA+'/
                          data
                                                    idc_command(6)
                                                                            /'READ DATA W/O HEADER CHECK+'/
                          data
                                                    idc_command(7)
0358
0359
                          logical*1
                                                   diagnostic_mode
0360
0361
0362
                          integer*4
                                                   libSextzv
                          integer*4
                                                   data_check_and_opi_bits
data_late_and_opi_bits
0363
                          integer+4
0364
0365
                          integer*4
                                                   sector_count
                                                   ecc_status_bits
rl02_status_bits
                          integer*4
0366
0367
                          integer*4
                                      virlo2 status_bits(0:7)
virlo2 status_bits(0) /'LOAD (1)
virlo2 status_bits(1) /'SPIN (1)
virlo2 status_bits(2) /'BRUSH
virlo2 status_bits(3) /'LOAD (1)
virlo2 status_bits(4) /'SEEK (1)
virlo2 status_bits(5) /'SEEK (1)
virlo2 status_bits(6) /'UNLOAD (1)
virlo2 status_bits(7) /'SPIN (1)
0368
0369
0370
0371
0372
0373
0376
0376
0378
0379
                          character*20
                                                                            /'LOAD STATE*'/
/'SPIN UP*'/
                          data
                          data
                                                                             /'BRUSH CYCLE+'/
/'LOAD HEADS+'/
/'SEEK TRACK COUNTING+'/
                          data
                          data
                          data
                                                                             /'SEEK LINEAR MODE+'/
                          data
                                                                             /'UNLOAD HEADS+'/
                          data
                                                                             /'SPIN DOWN+'/
                          data
                                                    device_function
                          integer*4
                                                   device_type
                          integer*4
0380
                          integer*4
                                                   sector
0381
                          integer*4
                                                   cylinder
0382
0383
                          integer*4
                                                   tag
                          integer*4
                                                   head
0384
0385
                                                   v1dar(0:1)
                          character*11
0386
                          data
                                                   v1dar(0)
                                                                             /'MARKER+'/
```

DQD

AP

ARR

LAB

FUN

```
DQDISKS
                                                                                16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
0387
0388
0389
0390
0391
0392
0393
                    data
                                        v1dar(1)
                                                            /'GET STATUS+'/
                                        v2dar(3:3)
v2dar(3)
                    character*6
                                                            /'RESET+'/
                    data
                                        v4dar(2:2,0:1)
v4dar(2,0)
v4dar(2,1)
                    character*8
                                                            /'REVERSE*'/
                    data
0394
0395
                                                            /'FORWARD+'/
                    data
0396
                    character*18
                                        v6dar(4:4,0:1)
0397
0398
0399
0400
                                        v6dar(4.0)
                                                            /'SELECT LOWER HEAD+'/
                    data
                                        v6dar(4,1)
                                                            /'SELECT UPPER HEAD+'/
                    data
                    character*15
                                        v7dar(6:6)
0401
                    data
                                        v7dar(6)
                                                            /'RETURN-TO-ZERO+'/
0402
0403
0404
                    call frctof (lun)
0405
0406
                    call dhead1 (lun, 'RB730')
0407
0408
                    diagnostic_mode = .false.
0409
0410
                    if (lib$extzv(25,1,control_status_register) .eq. 1)
0411
                    1 diagnostic_mode = .true.
0412
0413
                    device_function = libSextzv (1,3,control_status_register)
0414
0415
                    device_type = lib$extzv (26,1,control_status_register)
0416
0417
                    call linchk (lun,2)
0418
0419
                    write(lun,5) 'RB CSR',control_status_register
format(/' ',t8,a,t24,z8.8)
0420
          5
0421
0422
0423
0424
0425
0426
0427
0430
0431
0432
0433
                    if (.not. diagnostic_mode) then
                    call output (lun,control_status_register,v1csr,0,0,0,'0')
                    call linchk (lun,1)
                    if (lib$extzv(29,1,control_status_register) .eq. 1) then
                    write(lun,10) 'R80 WRITE FORMAT FUNCTION'
format(' ',t40,a)
          10
                    else
0434
                    idc_function = lib$extzv(1,3,control_status_register)
0436
0437
0438
0439
                    write(lun,15) idc_command(idc_function)
format(' ',t40,a<compressc (idc_command(idc_function))>)
          15
                    endif
0440
                    call output (lun,control_status_register,v2csr,6,6,7,'0')
0441
0442
                    call linchk (lun,1)
```

5

Page

VAX-11 FORTRAN V3.4-56

DISK\$VMSMASTER:[ERF.SRC]DQDISKS.FOR:1

COM

COP

COP R

```
DQDISKS
                                                                             16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
                   write(lun,20) 'DRIVE #',lib$extzv(8,2,control_status_register),
1 '. SELECTED'
format(' ',t40,a,i1.1,a)
0444
0446
         50
0447
0448
0449
0451
0452
0453
                   call output (lun,control_status_register,v3csr,10,10,10,'0')
                   data_check_and_opi_bits = lib$extzv(10,2,control_status_register)
                   1 data_check_and_opi_bits .eq. 2
                     .or.
0455
                     data_check_and_opi_bits .eq. 3
                   1) then
0457
0458
                   call linchk (lun,1)
0459
                   endif
0460
0461
                   if (data_check_and_opi_bits .eq. 2) then
0462
                   write(lun,25) 'DATA CHECK ERROR' format(' ',t40,a)
0463
         25
0464
0465
0466
                   else if (data_check_and_opi_bits .eq. 3) then
0467
0468
                   write(lun,25) 'HEADER CRC ERROR'
0469
0470
0471
                   endif
                   data_late_and_opi_bits = libSextzv(10,3,control_status_register)
0472
0473
0474
                   1 data_late_and_opi_bits .eq. 4
0475
0476
                     data_late_and_opi_bits .eq. 5
0477
                   1) then
0478
0479
0480
                   call linchk (lun,1)
                   endif
0481
0482
                   if (data_late_and_opi_bits .eq. 4) then
0483
0484
0485
0486
0487
0488
                   write(lun.25) 'DATA LATE'
                   else if (data_late_and_opi_bits .eq. 5) then
                   write(lun,25) 'HEADER NOT FOUND'
0489
                   endif
0490
0491
                   call output (lun,control_status_register,v4csr,13,13,15,'0')
0492
                   do 35, i = 16, 19
0494
0495
                   if (libSextzv(i,1,control_status_register) .eq. 1) then
0496
0497
                   call linchk (lun,1)
0498
                   write(lun,30) 'ATTENTION DRIVE #',i-16.'.' format(' ',t40,a,i1.1,a)
0499
0500
          30
```

**

VAX-11 FORTRAN V3.4-56 PDISK\$VMSMASTER: [ERF.SRC]DQDISKS.FOR;1

Page

VAX-11 FORTRAN V3.4-56

DISK\$VMSMASTER: [ERF.SRC]DQDISKS.FOR: 1

```
DODISKS
                                                                               16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
0501
0502
0503
                    endif
          35
                    continue
0504
0505
                    if (lib$extzv (26,1,control_status_register) .eq. 1) then
0506
0507
0508
                    ecc_status_bits = lib$extzv (20,2,control_status_register)
0509
                    if (ecc_status_bits .ne. 0) then
0510
0511
                    call linchk (lun.1)
0512
0513
                    if (ecc_status_bits .eq. 1) then
0514
                    write(lun,40) 'DATA ERROR' format(' ',t40,a)
0515
0516
          40
0517
0518
0519
0520
0521
0523
0523
0525
0526
0527
                    else if (ecc_status_bits .eq. 2) then
                    write(lun_40) 'HARD ERROR'
                    else if (ecc_status_bits .eq. 3) then
                    write(lun,40) 'CORRECTABLE ERROR'
                    endif
                    endif
                    endif
0528
0529
0530
                   call output (lun,control_status_register,v5csr,22,22,24,'0')
0531
0532
0533
                    if (lib$extzv (26,1,control_status_register) .eq. 1) then
                    call output (lun,control_status_register,v6csr,26,26,28,'0')
0534
0535
                    endif
                   else
0536
0537
0538
0539
0540
                   call linchk (lun,1)
                    write(lun,40) 'DIAGNOSTIC MODE'
                    endif
0541
0542
0543
                    call linchk (lun,1)
0544
0545
0546
0547
                    write(lun,45) 'RB_BAR',bus_address_register
          45
                    format(' '
                                ,t8,a,t24,z8.8)
                    if (.not. diagnostic_mode) then
0548
0549
0550
0551
                    if (
                      device_function .eq. 1
0552
0553
                      device_function .eq. 5
0554
                      device_function .eq. 6
0555
0556
                      device_function .eq. 7
0557
                      ) then
```

```
0558
0559
                  call_calc_map (lun,16,bus_address_register,bus_address_register)
0560
                  endif
0561
                  endif
0562
                  call linchk (lun,1)
0564
0565
                  write(lun,45) 'RB BCR',byte_control_register
0566
0567
0568
0569
0570
                  call linchk (lun,1)
                  write(lun,45) 'RB DAR',disk_address_register
0571
0572
                  if (.not. diagnostic_mode) then
0573
0574
                    device_function .eq. 1
0575
                     or.
0576
                    device_function .eq. 5
0577
                     .or.
0578
                    device_function .eq. 6
0579
0580
                    device_function .eq. 7
0581
                    ) then
0582
0583
                  if (device_type .eq. 0) then
0584
0585
                  sector = libSextzv (0,6,disk_address_register)
0586
0587
                  cylinder = lib$extzv (7,9,disk_address_register)
0588
0589
                  else if (device_type .eq. 1) then
0590
0591
                  sector = lib$extzv (0,5,disk_address_register)
0592
0593
                  cylinder = lib$extzv (9,10,disk_address_register)
0594
0595
0596
0597
                  endif
                  call linchk (lun,2)
0598
0599
                  write(lun,46) sector.cylinder format(',t40,'SECTOR #'.i<c
         46
                             ,t40,'SECTOR'#',i<compress4 (sector)>,'.',/,
0600
                  1 t40, 'CYLINDER #', i < compress4 (cylinder) >, '.')
0601
0602
                  else if (device_function .eq. 2) then
0603
0604
                  if (device_type .eq. 0) then
0605
0606
                  call output (lun,disk_address_register,v1dar,0,0,1,'0')
0607
0608
                  call output (lun, disk_address_register, v2dar, 3, 3, 3, '0')
0609
                  endif
0610
0611
                  else if (device_function .eq. 3) then
0612
0613
                  if (device_type .eq. 0) then
0614
```

```
16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
DQDISKS
                                                                                                  VAX-11 FORTRAN V3.4-56
                                                                                                  DISKSVMSMASTER: [ERF.SRC]DQDISKS.FOR: 1
0615
                  call output (lun, disk_address_register, v1dar, 0, 0, 1, '0')
0616
0617
                  call output (lun, disk_address_register, v4dar, 2, 2, 2, '2')
0618
0619
                  call output (lun, disk_address_register, v2dar, 3, 3, 3, '0')
0650
0623
0623
                  call output (lun, disk_address_register, v6dar, 4, 4, 4, 4, '2')
                  cylinder = lib$extzv (7,9,disk_address_register)
0624
0625
                  call linchk (lun,1)
0626
0627
0628
0629
0630
                  write(lun,47) cylinder
         47
                  format(' ',t40,i<compress4 (cylinder)>,'. CYLINDER(S) TO MOVE')
                  else if (device_type .eq. 1) then
0631
0632
                  tag = lib$extzv (13,3,disk_address_register)
0633
0634
                  call linchk (lun,1)
0635
0636
                  if (tag .eq. 1) then
0637
0638
                  cylinder = lib$extzv (0,10,disk_address_register)
0639
0640
                  write(lun,48) 'CYLINDER #',cylinder
         48
0641
                  format(' ',t40,a,i<compress4 (cylinder)>,'. SELECTED')
0642
                  else if (tag .eq. 2) then
0644
0645
                  head = lib$extzv (0,4,disk_address_register)
0646
0647
                  write(lun,48) 'HEAD #',head
0648
0649
                  else if (tag .eq. 4) then
0650
9651
                  call_output (lun,disk_address_register,v7dar,6,6,6,'0')
0652
                  endif
0653
                  endif
0654
                  endif
0655
                  endif
0656
0657
                  call linchk (lun,1)
0658
                  write(lun,50) 'RB MPR',multi_purpose_register
format(' ',t8,a,t24,z8.8)
0659
         50
0660
0661
0662
                  if (.not. diagnostic_mode) then
0663
0664
                  if (lib$extzv (26,1,control_status_register) .eq. 1) then
0665
0666
                  sector_count = lib$extzv (0,5,multi_purpose_register)
0667
0668
                  call linchk (lun,1)
0669
                  write(lun,55) 'SECTOR COUNT ',sector_count.'.'
0670
0671
         55
                  format(' ',t40,a,i<compress4 (sector_count)>,a)
```

```
7
                                                                            16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
DQDISKS
                                                                                                         VAX-11 FORTRAN V3.4-56
                                                                                                                                                   Page 10
                                                                                                         DISK$VMSMASTER:[ERF.SRC]DQDISKS.FOR:1
0672
0673
0674
0675
0676
0677
0678
                   call cutput (lun,multi_purpose_register,v1r80_mpr,8,8,13,'0')
                   else
                   rlO2_status_bits = lib$extzv (0,3,multi_purpose_register)
                   call linchk (lun,1)
                   write(lun,60) v1rl02_status_bits(rl02_status_bits)
format(' ',t40,a<compressc Tv1rl02_status_bits(rl02_status_bits))>)
0680
0681
         60
0682
0683
                   call output (lun,multi_purpose_register,v1rl02_mpr,3,3,5,'0')
0684
0685
                   call linchk (lun,1)
0686
0687
                   if (lib$extzv (6,1,multi_purpose_register) .eq. 1) then
0688
0689
                   write(lun,65) 'LOWER HEAD SELECTED'
0690
         65
                   format('
                               ,t40,a)
0691
0692
0693
                   else
                   write(lun,65) 'UPPER HEAD SELECTED'
0694
                   endif
0695
0696
                   call output (lun,multi_purpose_register,v2rl02_mpr,8,8,15,'0')
0697
                   endif
0698
                   endif
0699
0700
                   call linchk (lun,2)
0701
0702
                   write(lun,70) 'RB ECC1',ecc_position_register,
1 'RB ECC2',ecc_pattern_register
0703
                   format(' ',t8,a,t24,z8.8,/,t8,a,t24,z8.8)
         70
0704
0705
0706
0707
                   1 (device_function .eq. 1
0708
0709
                     device_function .eq. 5
0710
0711
                     device_function .eq. 6
0712
0713
                     device_function .eq. 7)
0714
                     .and.
0715
                     emb$w_hd_entry .ne. 98
0716
                   1 ) then
0717
0718
                   call uba_datapath (lun, iand(data_path_number, '0000007f'x),
0719
                   1 data_path_register)
0720
0721
0722
0723
0724
0725
                   call calc_map2 (16,bus_address_register,bus_address_register,field)
                   call uba_mapping (lun,field,final_map_register)
0726
0727
0728
                   1 lib$extzv (16,16,emb$l_dv_iosb1) .gt. 512
                     .and.
                   1 field .ne. 0
```

VAX-11 FORTRAN V3.4-56 Page 11 DISK\$VMSMASTER:[ERF.SRC]DQDISKS.FOR;1

```
K 7
16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
DQDISKS
0729
07312
07334
07334
07336
0736
07441
0745
                   1) then
                   call_uba_mapping (lun,(field-1),previous_map_register)
                   call vecmapreg (lun, vec$l_mapreg)
                   call linchk (lun,1)
                   write(lun,75)
format(' ',:)
         75
                   if (emb$w_hd_entry .ne. 98) then
                   call ucb$b_ertcnt (lun,emb$b_dv_ertcnt)
0746
                   call ucb$b_ertmax (lun,emb$b_dv_ertmax)
0747
0748
                   endif
0749
0750
                   call orb$l_owner (lun,emb$l_dv_ownuic)
0751
0752
0753
                   call ucb$i_char (lun,emb$l_dv_char)
                   call ucb$w_sts (lun,emb$w_dv_sts)
0754
0755
                   call ucb$l_opcnt (lun,emb$l_dv_opcnt)
0756
0757
0758
                   call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0759
                   if (emb$w_hd_entry .ne. 98) then
0760
                   call ucb$l_media (lun,emb$l_dv_media)
0761
0762
0763
                   call linchk (lun,1)
0764
0765
                   write(lun,75)
0766
0767
                   call dqdisks_qio (lun,emb$w_dv_func)
0768
0769
0770
                   call irp$w_bcnt (lun,emb$w_dv_bcnt)
0771
0772
0773
                   call irp$w_boff (lun,emb$w_dv_boff)
                   call irp$l_pid (lun,emb$l_dv_rqpid)
0774
0775
                   call_irp$q_iosb (lun,emb$l_dv_iosb1)
0776
0777
0778
                   endif
                   return
0779
                   end
```

```
16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
DQDISKS
                                                                                                                                                                                                                                                                                               VAX-11 FORTRAN V3.4-56
DISK$VMSMASTER: [ERF.SRC]DQDISKS.FOR; 1
                                                                                                                                                                                                                                                                                                                                                                                                                       Page 12
PROGRAM SECTIONS
            Name
                                                                                                                                                                 Attributes
                                                                                                                                       Bytes
      O SCODE
                                                                                                                                          2861
634
2972
                                                                                                                                                                 PIC CON REL LCL
                                                                                                                                                                                                                           SHR
                                                                                                                                                                                                                                                                   RD NOWRT LONG
                                                                                                                                                                                                                                            EXE
       1 SPDATA
                                                                                                                                                                                                                           SHR NOEXE
                                                                                                                                                                 PIC CON REL LCL
                                                                                                                                                                                                                                                                   RD NOWRT LONG
       2 $LOCAL
3 EMB
                                                                                                                                                                 PIC CON REL LCL NOSHR NOEXE
                                                                                                                                                                                                                                                                   RD
                                                                                                                                                                                                                                                                                   WRT LONG
                                                                                                                                             512
                                                                                                                                                                 PIC OVR REL GBL
                                                                                                                                                                                                                            SHR NOEXE
                                                                                                                                                                                                                                                                   RD
                                                                                                                                                                                                                                                                                   WRT LONG
                                                                                                                                          6979
            Total Space Allocated
ENTRY POINTS
            Address Type Name
      0-00000000
                                                              DODISKS
VARIABLES
                                      Name

BUS_ADDRESS_REGISTER
CONTROL_STATUS_REGISTER
DATA_CHECK_AND_OPI_BITS
ADATA_PATH_RUMBER
DEVICE_FUNCTION
DIAGNOSTIC_MODE
ACC_PATTERN_REGISTER
ACC_STATUS_BITS
BUS_ADDRESS_REGISTER
ADATA_PATH_RUMBER
ADATA_PATH_RUMBER
ADATA_PATH_RUMBER
ADATA_PATH_RUMBER
ADATA_PATH_RUMBER
ADATA_PATH_RUMBER
ADATA_PATH_RUMBER
ADATA_PATH_REGISTER
ADATA_PATH_RUMBER
ADATA_PATH_REGISTER
ADATA_PATH_RUMBER
ADATA_PATH_REGISTER
ADATA_PATH_RUMBER
ADATA_PATH_REGISTER
ADATA_PATH_RUMBER
A
            Address Type Name
                                                                                                                                                                                    Address Type Name
       3-00000056
                                                                                                                                                                                                                                     BYTE_CONTROL_REGISTER CYLINDER
                                                                                                                                                                               3-0000005A
        3-00000052
                                                                                                                                                                               2-00000424
                                                                                                                                                                                                                     1+4
                                                                                                                                                                                                                                     DATA_LATE_AND_OPI_BITS
DATA_PATH_REGISTER
DEVICE_TYPE
        2-00000404
                                                                                                                                                                                2-00000408
                                                                                                                                                                                                                     1+4
                                                                                                                                                                              3-00000072
       3-0000000E
                                                                                                                                                                                                                     I+4
                                                                                                                                                                                                                     I+4
I+4
        2-00000418
                                                                                                                                                                               2-0000041C
                                                                                                                                                                              3-00000055
                                                                                                                                                                                                                                     DISK ADDRESS REGISTER ECC POSITION REGISTER
        2-000003FF
       3-0000006A
                                                                                                                                                                               3-00000066
                                                                                                                                                                                                                     1+4
                                                                                                                                                                                                                                    ECC_POSITION_REGEMB$B_DV_CLASS
EMB$B_DV_ERTMAX
EMB$B_DV_SLAVE
EMB$L_DV_CHAR
EMB$L_DV_IOSB2
EMB$L_DV_NUMREG
EMB$L_DV_OWNUIC
EMB$L_DV_OWNUIC
EMB$L_HD_SID
EMB$W_DV_ERRCNT
EMB$W_DV_ERRCNT
EMB$W_DV_STS
FMB$W_HD_FNTRY
        2-00000410
                                                                                                                                                                               3-0000001C
                                                                                                                                                                                                                     L+1
        3-00000010
                                                                                                                                                                              3-00000011
                                                                                                                                                                                                                     L+1
       3-0000003E
                                                                                                                                                                                                                     L*1
I*4
I*4
                                                                                                                                                                              3-0000003A
       3-0000001D
                                                                                                                                                                              3-00000036
3-00000016
        3-00000012
        3-00000026
                                                                                                                                                                              3-0000004E
3-00000032
                                                                                                                                                                                                                     1+4
       3-0000002E
                                                                                                                                                                                                                     1+4
       3-0000001E
                                                                                                                                                                              3-00000000
                                                                                                                                                                                                                     1+4
                                                                                                                                                                             3-00000024
3-0000002C
3-0000001A
3-00000004
       3-0000003F
        3-00000022
        3-0000003C
       3-0000002A
                                                                                                                                                                                                                                     EMB$W_HD_ENTRY
                                                                                                                                                                             2-00000400
2-000042C
2-00000430
3-00000062
2-00000414
2-0000040C
3-0000007E
                                                                                                                                                                                                                                     FIELD'
        3-0000000E
                                                                                                                                                                                                                      1+4
        3-00000076
                                                                                                                                                                                                                      1+4
                                                                                                                                                                                                                                      HEAD
                                                                                                                                                                                                                                     IDC FUNCTION
MULTI PURPOSE REGISTER
RLO2 STATUS BITS
SECTOR COUNT
        2-00000434
                                                                                                                                                                                                                     I+4
    AP-00000004a L+1
                                                                                                                                                                                                                     1 *4
                                                              LUN
                                                             PREVIOUS_MAP_REGISTER
                                                                                                                                                                                                                    I • 4
I • 4
       3-000007A I+4
      2-00000420 2-0000428
                                              1 * 4
                                                              SECTOR
                                             1 * 4
                                                              TAG
                                                                                                                                                                                                                                      VEC$L_MAPREG
```

PRO

ENT

0

VAR

AP

ARR

DQDISKS	16-Sep-1984 5-Sep-1984	00:02:07 13:52:37	VAX-11 FORTRAN V3.4-50 DISK\$VMSMASTER:[ERF.SF	6 Page 13 RC]DQDISKS.FOR;1
ARRAYS				
Address Type Name	Bytes Dimensions			
3-0000000 L*1 EMB 3-00000052 I*4 EMB\$L_DV_REGSAV 3-00000006 I*4 EMB\$Q_HD_TIME 2-00000228 CHAR IDC_COMMAND 2-000003A0 CHAR V1C\$R 2-000001D4 CHAR V1RB0_MPR 2-0000011B CHAR V1RL02_MPR 2-00000300 CHAR V1RL02_STATUS_BITS 2-000003B6 CHAR V2C\$R 2-000003B6 CHAR V2DAR 2-0000013C CHAR V2C\$R 2-0000013C CHAR V4C\$R 2-00000043 CHAR V4C\$R 2-0000007F CHAR V4C\$R 2-0000007F CHAR V5C\$R 2-00000061 CHAR V6C\$R 2-000003CC CHAR V6DAR 2-000003FO CHAR V7DAR	512 (0:511) 420 (0:104) 8 (2) 216 (0:7) 12 (0:0) 22 (0:1) 84 (8:13) 33 (3:5) 160 (0:7) 34 (6:7) 6 (3.7) 152 (8:15) 21 (10:10) 60 (13:15) 16 (2:2, 0:1) 66 (22:24) 90 (26:28) 36 (4:4, 0:1) 15 (6:6)			
LABELS				
Address Label Address Label	Address Label Addres	s Label	Address Label	Address Label
1-00000176 5' 1-00000183 10' ** 35 1-00000183 40' 1-00000233 50' 1-0000023F 55'	1-0000018A 15' 1-000001 1-000001BA 45' 1-000001 1-0000024D 60' 1-000002	C6 46'	1-000001A1 25' 1-000001F7 47' 1-00000260 70'	1-000001A8 30' 1-0000021A 48' 1-00000275 75'
FUNCTIONS AND SUBROUTINES REFERENCED	•			
Type Name Type Name	Type Name Type Na	me	Type Name	Type Name
CALC_MAP FRCTOF IRP\$[_PID LINCHK ORB\$L_OWNER UCB\$B_ERTMAX UCB\$L_CHAR VECMAPREG	IRP\$Q_IOSB IR OUTPUT UB	MPRESSC PSW_BCNT IA_DATAPATI BSL_OPCNT	DHEAD1 IRP\$W_BOFF H UBA_MAPPING UCB\$W_ERRCNT	DQDISKS_Q10 I+4 LIB\$EXTZV UCB\$B_ERTCNT UCB\$W_STS

LAB

FUN

T

```
N 7
16-Sep-1984 00:02:07
5-Sep-1984 13:52:37
```

Subroutine DQDISKS_QIO (lun,emb\$w_dv_func) include 'src\$:giocommon.for /nolist' byte lun integer*2 emb\$w_dv_func integer*4 qiocode(0:1,0:63) if (qiocode(0,0).eq. 0) then $qiocode(1,00) = %loc(io\$_nop)$ qiocode(1,02) = %loc(io\$_seek) qiocode(1,03) = %loc(io\$_recal) qiocode(1,04) = %loc(io\$_drvclr) qiocode(1,08) = %loc(io\$_packack) qiocode(1,10) = %loc(io\$_writecheck) qiocode(1,11) = %loc(io\$_writepblk) qiocode(1,12) = %loc(io\$_readpblk) qiocode(1,14) = %loc(io\$_readhead) qiocode(1,26) = %loc(io\$_setchar) qiocode(1,27) = %loc(io\$_sensechar) qiocode(1,32) = %loc(io\$_writelblk) qiocode(1,33) = %loc(io%_readlblk) qiocode(1,35) = %loc(io\$_setmode)

qiocode(1,39) = %loc(io\$_sensemode)

qiocode(1,48) = %loc(io\$_writevblk)

```
DQDISKS_Q10
                   qiocode(1,49) = %loc(io$_readvblk)
                   qiocode(1,50) = %loc(io$_access)
                   qiocode(1,51) = %loc(io$_create)
                   qiocode(1,52) = %loc(io$_deaccess)
                   qiocode(1,53) = %loc(io$_delete)
                   qiocode(1,54) = %loc(io$_modify)
                   qiocode(1,56) = %loc(io$_acpcontrol)
                   qiocode(1,57) = %loc(io$_mount)
                   do 10,i = 0.63
0339
                   qiocode(0,i) = 33
0340
0341
                   if (qiocode(1,i) .eq. 0) then
                   qiocode(1,i) = %loc(qio_string)
endif
0343
0344
0345
0346
0347
0348
0349
          10
                   continue
                   endif
                   call irp$w_func (lun,emb$w_dv_func,
1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))
0351
0352
0353
0354
                   return
                   end
```

DR'

Page 15

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER: [ERF.SRC]DQDISKS.FOR; 1

03;

PR(

EN1

VAF

AF

```
DQDISKS_Q:O
PROGRAM SECTIONS
                      Name
                                                                                                                                                                                                                    Bytes Attributes
                                                                                                                                                                255 PIC CON REL LCL SHR EXE
8 PIC CON REL LCL SHR NOEXE
548 PIC CON REL LCL NOSHR NOEXE
1247 PIC OVR REL GBL SHR NOEXE
           O SCODE
                                                                                                                                                                                                                                                                                                                                                                                                                           RD NOWRT LONG
                                                                                                                                                                                                                                                                                                                                                                                                                         RD NOWRT LONG
RD WRT LONG
RD WRT LONG
            1 SPDATA
            2 SLOCAL
3 QIOCOMMON
                     Total Space Allocated 2058
  ENTRY POINTS
                 Address Type Name
                                                                          DQDISKS_Q10
    0-00000000
 VARIABLES
Address Type Name

Address Type Name

Address Type Name

Ap-000000083 I*2 EMB$W DV FUNC
3-00000340 CHAR I0$ ABORT
3-00000340 CHAR I0$ ACCESS
3-00000297 CHAR I0$ ACCESS
3-00000287 CHAR I0$ ACCESS
3-00000285 CHAR I0$ DEACCESS
3-00000385 CHAR I0$ DEACCESS
3-00000385 CHAR I0$ DEACCESS
3-00000260 CHAR I0$ DEACCESS
3-00000408 CHAR I0$ DEACCESS
3-00000276 CHAR I0$ DOWN CHAR I0$ DEACCESS
3-00000276 CHAR I0$ DOWN CHAR I0$ DEACCESS
3-00000077 CHAR I0$ DOWN CHAR I0$ DEACCESS
3-00000078 CHAR I0$ DOWN CHAR I0$ DOWN CHAR I0$ DEACCESS
3-00000076 CHAR I0$ DOWN CHAR I0$ DOWN CHAR I0$ DOWN CHAR I0$ DEACCESS
3-00000077 CHAR I0$ DOWN CHAR I0$ D
                                                                                                                                                                                                                                                          Address Type Name
                      Address Type Name
```

ARR

LAB

FUN

COM

COP

D 8 16-Sep-1984 00:02:07 5-Sep-1984 13:52:37

VAX-11 FORTRAN V3.4-56 Page 17 DISK\$VMSMASTER:[ERF.SRC]DQDISKS.FOR;1

3-0000017E CHAR IOS_WRITETRACKD 3-00000448 CHAR IOS_WRITEWTHBUF AP-000000040 L+1 LUN 3-00000326 CHAR IOS_WRITEVBLK 3-00000257 CHAR IOS_WRITMKR 3-000004A1 CHAR QIO_STRING

ARRAYS

Address Type Name

Bytes Dimensions

2-00000000 I+4 Q10CODE

512 (0:1, 0:63)

LABELS

Address Label

10

FUNCTIONS AND SUBROUTINES REFERENCED

Type Name

Type Name

IRP\$W_FUNC

1+4 LIBSEXTZV

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:DQDISKS/OBJ=OBJ\$:DQDISKS MSRC\$:DQDISKS

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

10.94 seconds 25.02 seconds 269 Run Time: **Elapsed Time:** Page faults:

Dynamic Memory:

248 pages

0147 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

1 miles

	Fig. 19 (1997) Annual Control of	The second secon	CLASSIFY LIS	Great sustain		The second secon	Specification of the second se	2 131 and 191	Villed 191	E Tilenen	Garas, mannes,	General August	B. THE SECOND	TOTAL STATE OF THE	Grant dance of
E III III E	Anna Gana Anna Anna Anna Anna Anna Anna	The state of the s	TOTAL CONTROL OF THE PARTY OF T	The second	B Jilled 198	THE SAME	2 111 ran 191	RIPE BILLS			the second	The State of		Sant Santa-	
	The second secon	in Tinisaisa	Section 1	SEC. U.T.	III Base 16	I III BE	Man space to make make make make make make make make	DR250 LIS	SEE 122	Wester Australia	Charte manage.	dans stands		I III Michael Marketter and Michael Marketter and Michael Mi	Interior
The second secon	Section Control of Con	Mathematical Mathe	B THE SALE	CSTRING . LIS	DHEADS LIS	Mary 115 Market (America) Control Control	BESSEL	Market Administration (Market Administration	DR280 LIS	2 THE RESIDENCE	Harris Andrews	Mentional Park Manager 198 (1984) (19	these manner.	Geren manner	One of the last to
	Market Ma		Total Residence Control Reside	WE THE PROPERTY OF THE PROPERT	Marin	Control of the contro	DR11W LIS	Harrison	Shorts manuful and the same of	Manual State of State	Service matrices		B SHIP AND THE	SCHOOL STATE	there agents——
The first matters	Section And Section Assessment Control of the Contr	Billion	COMPRESS LIS	With the second	than many	** ** ** ** ** ** ** ** ** ** ** ** **	Maria	Sign Matter Address.	the constant of the constant o	DTAILS LIS	Martin Ma		Same and the same	Services Servic	them annual and a second and a
The second secon	The second secon	Timed via	TECHNOLOGY PARTY STATES OF THE PARTY STATES OF	DECODECC LIS	Company Agents	Magnetic Authorities of the Control	I L II WAR CICIO- CICIO	GOTON AMERICAN	I DE DESCRIPTION DE LA CONTRACTION DEL CONTRACTION DE LA CONTRACTI	Gentle Marie Communication of the Communication of	Grand Andreas	Control Section Control Sectin Control Section Control Section Control Section Control Section	State Cam	Enter Gamerican	MINE DINE
	The state of the s	William Removed Proportion State Part		TOTAL CONTROL OF THE PROPERTY	MATTER MATTER AND	September 1997 Septem	The second secon	NATIONAL MARKET PROPERTY OF THE PROPERTY OF TH	Maria nameta Simple and nameta Simpl	Garter manters.	The state of the s	The second secon	DUMPREG LIS	Time agree	DUTUDRIVR LIS
	TOTAL SERVICES	CALCMAP LIS	Ment dahara		Matter Authorities	And the second s	Martin Ma	Marine Authors - VESTINA - VEST	Grand Authority			Sign and American Control of the Con	Secretarian	ms m	Service Management (Service Management (Servic
The second secon		CONTROL OF THE PROPERTY OF THE			P IIII da 191			- WEARDOOM, ASSESSMENT PROPERTY OF THE PROPERT			Unite automate.	Control States and tree	Santa particular	DUP3271 LIS	The second secon
SCHOOLS AMERICAN			NAMES OF THE PROPERTY OF THE P		Section same	1 111 de 119	The Dir	10/20/20/20/20/20/20/20/20/20/20/20/20/20					B INTERNATIONAL PROPERTY OF THE PROPERTY OF TH	Sense Asserted	When manners and a control of the co
		Series and a serie	- CRYPTK LIS		DQDISKS LIS			TOTAL STATE OF THE						The second construction of the second constructi	2 11114a 191
The same state of the same sta		PROPERTY AND ASSESSED FOR THE PROPERTY ASSESSED FOR THE PROPERTY ASSESSED FOR THE PROPERTY ASSESSED.	TO A THE PROPERTY OF THE PROPE		STATE OF THE PROPERTY OF THE P								DUP11		